



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/800,122	03/12/2004	Nobuo Togahara	4142.70026	9096
24978	7590	11/26/2007		
GREER, BURNS & CRAIN 300 S WACKER DR 25TH FLOOR CHICAGO, IL 60606			EXAMINER LU, KUEN S	
			ART UNIT 2167	PAPER NUMBER
			MAIL DATE 11/26/2007	DELIVERY MODE PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No. 10/800,122	Applicant(s) TOGAHARA, NOBUO	
	Examiner Kuen S. Lu	Art Unit 2167	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 13 September 2007.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-21 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-21 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☒ All b) ☐ Some * c) ☐ None of:
1. ☒ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. The Action is responsive to Applicant's Amendment and Request for Continued Examination filed on September 13, 2007.

As necessitated by the Amendment, Examiner hereby withdraws rejections to claims 8-9, 15, 18 and 21 under 35 U.S.C. § 101.

2. Please note claims 1 and 2-21 are pending.

3. As to Applicant's Arguments/Remarks filed *September 13, 2007*, please see Examiner's response in "**Response to Arguments**"; following this Office Action for non-Final Rejection (hereafter "the Action"), shown next.

Claim Rejections - 35 USC § 101

4. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

4.1. As set forth in MPEP 2106 (II) (A):

The claimed invention as a whole must accomplish a practical application. That is, it must produce a "useful, concrete and tangible result." *State Street*, 149 F.3d at 1373, 47 USPQ2d at 1601-02. The purpose of this requirement is to limit patent protection to inventions that possess a certain level of "real world" value, as opposed to subject matter that represents nothing more than an idea or concept, or is simply a starting point for future investigation or research (*Brenner v. Manson*, 383 U.S. 519, 528-36, 148 USPQ 689, 693-96); *In re Ziegler*, 992, F.2d 1197, 1200-03, 26 USPQ2d 1600, 1603-06 (Fed. Cir. 1993)). Accordingly, a complete disclosure should contain some indication of the practical application for the claimed invention, i.e., why the applicant believes the claimed invention is useful.

Apart from the utility requirement of 35 U.S.C. 101, usefulness under the patent eligibility standard requires significant functionality to be present to satisfy the useful result aspect of the practical application requirement. See *Arrhythmia*, 958 F.2d at 1057, 22 USPQ2d at 1036. Merely claiming nonfunctional descriptive material stored in a computer-readable medium does not make the invention eligible for patenting. For example, a claim directed to a word processing file stored on a disk may satisfy the utility requirement of 35 U.S.C. 101 since the information stored may have some "real world" value. However, the mere fact that the claim may satisfy the utility requirement of 35 U.S.C. 101 does not mean that a useful result is achieved under the practical application requirement. The claimed invention as a whole must produce a "useful, concrete and tangible" result to have a practical application.

4.2. As set forth in MPEP 2106 (IV) (B) (1):

Claims to computer-related inventions that are clearly nonstatutory fall into the same general categories as nonstatutory claims in other arts, namely natural phenomena such as magnetism, and abstract ideas or laws of nature which constitute "descriptive material." Abstract ideas, *Warmerdam*, 33 F.3d at 1360, 31 USPQ2d at 1759, or the mere manipulation of abstract ideas, *Schrader*, 22 F.3d at 292-93, 30 USPQ2d at 1457-58, are not patentable. Descriptive material can be characterized as either "functional descriptive material" or "nonfunctional descriptive material." In this context, "functional descriptive material" consists of data structures and computer programs which impart functionality when employed as a computer component. (The definition of "data structure" is "a physical or logical relationship among data elements, designed to support specific data manipulation functions." The New IEEE Standard Dictionary of Electrical and Electronics Terms 308 (5th ed. 1993).) "Nonfunctional descriptive material" includes but is not limited to music, literary works and a compilation or mere arrangement of data. Both types of "descriptive material" are nonstatutory when claimed as descriptive material *per se*. *Warmerdam*, 33 F.3d at 1360, 31 USPQ2d at 1759. When functional descriptive material is recorded on some computer-readable medium it becomes structurally and functionally interrelated to the medium and will be statutory in most cases since use of technology permits the function of the descriptive material to be realized. Compare *In re Lowry*, 32 F.3d 1579, 1583-84, 32 USPQ2d 1031, 1035 (Fed. Cir. 1994) (claim to data structure stored on a computer readable medium that increases computer efficiency held statutory) and *Warmerdam*, 33 F.3d at 1360-61, 31 USPQ2d at 1759 (claim to computer having a specific data structure stored in memory held statutory product-by-process claim) with *Warmerdam*, 33 F.3d at 1361, 31 USPQ2d at 1760 (claim to a data structure *per se* held nonstatutory).

4.3. As set forth in MPEP 2106 (IV)(B)(1)(a):

Similarly, computer programs claimed as computer listings *per se*, *i.e.*, the descriptions or expressions of the programs, are not physical "things." They are neither computer components nor statutory processes, as they are not "acts" being performed. Such claimed computer programs do not define any structural and functional interrelationships between the computer program and other claimed elements of a computer which permit the computer program's functionality to be realized. In contrast, a claimed computer-readable medium encoded with a computer program is a computer element which defines structural and functional interrelationships between the computer program and the rest of the computer which permit the computer program's functionality to be realized, and is thus statutory. Accordingly, it is important to distinguish claims that define descriptive material *per se* from claims that define statutory inventions.

Products may be either machines, manufactures, or compositions of matter.

A machine is "a concrete thing, consisting of parts or of certain devices and combinations of devices." *Burr v. Duryee*, 68 U.S. (1 Wall.) 531, 570 (1863).

If a claim defines a useful machine or manufacture by identifying the physical structure of the machine or manufacture in terms of its hardware or hardware and software combination, it defines a statutory product. See, e.g., *Lowry*, 32 F.3d at 1583, 32 USPQ2d at 1034-35; *Warmerdam*, 33 F.3d at 1361-62, 31 USPQ2d at 1760.

Office personnel must treat each claim as a whole. The mere fact that a hardware element is recited in a claim does not necessarily limit the claim to a specific machine or manufacture. Cf. *In re Iwahashi*, 888 F.2d 1370, 1374-75, 12 USPQ2d 1908, 1911-12 (Fed. Cir. 1989), cited with approval in *Alappat*, 33 F.3d at 1544 n.24, 31 USPQ2d at 1558 n.24.

4.5. Claims 19 and 20 are rejected under 35 U.S.C. § 101 because the claimed invention is directed to non-statutory subject matter.

As per claims 19 and 20, the claims are apparatus comprising, respectively, (an analyzer, a checker and a register) and (an analyzer, a checker and a generator). The components of the apparatus seem to be software per se and it seems the subject matter claimed by the claims lacks of any hardware support. Therefore, the system does not fall in one of statutory categories of **useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof**. Consequentially, the claim is non-statutory.

Claim Rejections - 35 USC § 102

5. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless -

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

5.1. Claims 1-21 are rejected under 35 U.S.C. 102(b) as anticipated by Struts, The Jakarta Project, Apache Software Foundation, 2000-2001, hereafter "Struts").

As per claims 1, 16 and 19, Struts teaches "An method for supporting program development" (See Paragraph 1.2 where model-view-controller framework provides coupling between view and model for creating and maintaining applications), "A

program embodied on a computer readable medium, for causing a computer to execute an information processing, said program” (See Paragraphs 1.2 and 2.4 where model-view-controller framework provides coupling between view and model for creating and maintaining applications and beans are created on server’s memory) and “An information processing apparatus” (See Page 7, Paragraph Struts of Various Containers, where Struts containers include various servers), comprising:

“(an analyzer for) analyzing XML data corresponding to a form screen, and specifying a business class coded in object-oriented programming language, said business class corresponding to a tag included in said XML data” (See Page 43, Paragraph Element matching Patterns where XML document hierarchy is navigated and parsed and at Pages 73-74, Paragraph About the Form Tag where bean and its class is specified as a property of form tag and methods are provided to validate data in an object oriented programming environment);

“(a checker) referring to a business class manager in which items of said business classes, which are classes before compilation and to be created, are registered, and judging whether an item of the specified business class has not been registered in said business class manager” (See Page 43, Paragraph Element matching Patterns where processing rules for matching patterns are registered and patterns will be matched when corresponding element is parsed); and

“upon detecting that said item of the specified business class has not been registered in said business class manager, registering said item of the specified business class in said business class manager” and “(a register) registering said item of the specified

business class in said business class manager, if said item of the specified business class has not been registered in said business class manager” (See Page 16, Paragraph 2.3 where the Struts framework checks if session scope bean available and automatically creates a new one and adds to users session if the bean is not available).

As per claims 7, 17 and 20, Struts teaches “An method for supporting program development” (See Paragraph 1.2 where model-view-controller framework provides coupling between view and model for creating and maintaining applications), “A program embodied on a computer-readable medium, for causing a computer to execute an information processing, said program” (See Paragraphs 1.2 and 2.4 where model-view-controller framework provides coupling between view and model for creating and maintaining applications and beans are created on server’s memory) and “An information processing apparatus” (See Page 7, Paragraph Struts of Various Containers, where Struts containers include various servers), comprising:
“(an analyzer) analyzing XML data corresponding to a form screen, and specifying a business class coded in object-oriented programming language, said business class corresponding to a tag included in said XML data” (See Page 43, Paragraph Element matching Patterns where XML document hierarchy is navigated and parsed and at Pages 73-74, Paragraph About the Form Tag where bean and its class is specified as a property of form tag and methods are provided to validate data in an object oriented programming environment);

“(a checker) referring to a business class manager in which items of said business classes, which are classes before compilation and to be created, are registered, and judging whether an item of the specified business class has not been registered in said business class manager” (See Page 43, Paragraph Element matching Patterns where processing rules for matching patterns are registered and patterns will be matched when corresponding element is parsed); and

“upon detecting that said item of the specified business class has not been registered in said business class manager, generating template source program data for the specified business class” and “(a generator) generating template source program data for the specified business class, if said item of the specified business class has not been registered in said business class manager” (See Page 16, Paragraph 2.3 where the Struts framework checks if session scope bean available and automatically creates a new one and adds to users session if the bean is not available).

As per claims 8, 18 and 21, Struts teaches “An information processing method” (See Paragraph 1.2 where model-view-controller framework provides coupling between view and model for creating and maintaining applications), “A program embodied on a computer-readable medium, for causing a computer to execute an information processing,” (See Paragraphs 1.2 and 2.4 where model-view-controller framework provides coupling between view and model for creating and maintaining applications and beans are created on server’s memory) and “An information processing apparatus” (See Page 7, Paragraph Struts of Various Containers, where Struts containers include

various servers), comprising:

“(a first processor) obtaining XML data including a tag corresponding to data inputted or selected for a form screen, and specifying a business class, wherein said business class corresponds to said tag included in said XML data and is an executable program for carrying out a processing relating to said tag” (See Page 43, Paragraph Element matching Patterns where XML document hierarchy is navigated and parsed and at Pages 73-74, Paragraph About the Form Tag where bean and its class is specified as a property of form tag and methods are provided to validate data in an object oriented programming environment);

“(a second processor) executing the specified business class” (See Pages 12-13, Paragraph 1.5 where business objects are able to render themselves in XML, based on their current state at request time); and

“(a third processor) outputting data including a processing result by said specified and executed business class to an apparatus that displayed said form screen” (See Page 16, Paragraph 2.3 where the Struts framework checks if session scope bean available and automatically crates a new one and adds to users session if the bean is not available).

As per claim 2, Struts further teaches “The method as set forth in claim 1, further comprising: upon detecting that said item of the specified business class has not been registered in said business class manager, generating template source program data for the specified business class” (See Page 16, Paragraph 2.3 where the Struts framework

checks if session scope bean available and automatically creates a new one and adds to users session if the bean is not available).

As per claim 3, Struts teaches "The method as set forth in claim 1, further comprising: reading out an HTML file for said form screen, and generating XML data corresponding to said form screen according to a predetermined rule" (See Pages 12-13, Paragraph 1.5 where rendered HTML results in JSP page and XML file is represented as a collection of objects cached in controller servlet).

As per claim 4, Struts further teaches "The method as set forth in claim 1, wherein said analyzing and specifying comprises specifying a pre-processing class, a post-processing class and a form processing class, so as to correspond to a start tag or an end tag of said XML data corresponding to said form screen" (See Pages 9-10, Paragraph 1.3 where controller responds to HTTP request and directs client to JSP and tags populate fields from a form bean).

As per claim 5, Struts further teaches "The method as set forth in claim 3, further comprising: generating said HTML file for said form screen in response to an instruction of a user" (See Page 8, Paragraph 1.1 where JSP writes HTML).

As per claim 6, Struts teaches the information processing method as set forth in claim 1, further comprising:

“specifying a form item storing object by a tag included in said XML data corresponding to said form screen” (See Page 43, Paragraph Element matching Patterns where XML document hierarchy is navigated and parsed and at Pages 73-74, Paragraph About the Form Tag where bean and its class is specified as a property of form tag and methods are provided to validate data in an object oriented programming environment);

“referring to a form item storing object manager in which items of form item storing objects to be created are registered, and judging whether an item of the specified form item storing object has not been registered in said form item storing object manager” (See Page 43, Paragraph Element matching Patterns where processing rules for matching patterns are registered and patterns will be matched when corresponding element is parsed); and

“upon detecting that said item of the specified form item storing object has not been registered in said form item storing object manager, registering said item of the specified form item storing object into said form item storing object manager” (See Page 16, Paragraph 2.3 where the Struts framework checks if session scope bean available and automatically creates a new one and adds to users session if the bean is not available).

As per claim 9, Struts teaches “The information processing method as set forth in claim 8, wherein said obtaining and specifying comprises specifying a pre-processing class, a post-processing class, and a form processing class that correspond to a form of said form screen and a tag included in said XML data” (See Page 43, Paragraph

Element matching Patterns where XML document hierarchy is navigated and parsed and at Pages 73-74, Paragraph About the Form Tag where bean and its class is specified as a property of form tag and methods are provided to validate data in an object oriented programming environment).

As per claim 10, Struts teaches the information processing method as set forth in claim 8, further comprising:

"outputting said data inputted or selected for said form screen to a form item storing object that is defined in advance and is loaded into a memory" (See Page 118, Paragraph 1.4 where the rendered output of objects in JSP by using action tag);

"storing said data inputted or selected for said form screen into said memory by said form item storing object" (See Page 120 Paragraph 2.4 where system state beans are created on memory); and

"performing a processing by exchanging data between the called business class and said form item storing object" (See Page 120 Paragraph 2.4 where system state beans are created on and removed from memory by application processing).

As per claim 11, Struts teaches the information processing method as set forth in claim 8, further comprising:

"outputting data to an interclass interface object that is defined in advance and is loaded into said memory, by a first business class, in order to transfer said data from said first business class to said second business class" (See Page 118, Paragraph 1.4 where

the rendered output of objects in JSP by using action tag); and
“referring to said interclass interface object and receiving said data from said interclass interface object by said second business class” (See Page 43, Paragraph Element matching Patterns where processing rules for matching patterns are registered and patterns will be matched when corresponding element is parsed).

As per claim 12, Struts teaches the information processing method as set forth in claim 8, further comprising:

“receiving data inputted or selected for said form screen from an apparatus that displayed said form screen, and generating said XML data including said data inputted or selected for said form screen and corresponding tags” (See Page 43, Paragraph Element matching Patterns where XML document hierarchy is navigated and parsed and at Pages 73-74, Paragraph About the Form Tag where bean and its class is specified as a property of form tag and methods are provided to validate data in an object oriented programming environment).

As per claim 13, Struts further teaches “the information processing method as set forth in claim 8, wherein said outputting comprises:

“upon detecting that an output request is received from the executed business class, generating output XML data based on said processing result by said specified and executed compiled business class” (See Page 16, Paragraph 2.3 where the Struts framework checks if session scope bean available and automatically crates a new one

and adds to users session if the bean is not available); and
“outputting said output XML data to said apparatus that displayed said form
screen” (See Page 118, Paragraph 1.4 where the rendered output of objects in JSP by
using action tag).

As per claim 14, Struts teaches “The information processing method as set forth in
claim 8, further comprising: upon detecting that the specified business class does not
exist, generating and outputting error information” (See Page 16, Paragraph 2.3 where
the Struts framework checks if session scope bean available and automatically crates a
new one and adds to users session if the bean is not available).

As per claim 15, Struts further teaches “The information processing method as set
forth in claim 8, wherein each of the called business classes is configured so as to
complete a processing for the entire form relating to said form screen by a processing
executed by the called business classes without a program defining a processing
sequence” (See Page 13, Paragraph 1.6 where servlet is configured by defining a set of
mappings between an incoming class and business class).

References

6.1. The prior art made of record

U. Struts, The Jakarta Project, Apache Software Foundation, 2000-2001

6.2. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

- A. U.S. Patent Application 2003/0149934
- F. U.S. Patent Application 2002/0138582
- B. U.S. Patent Application 2004/0006527
- C. U.S. Patent Application 2003/0233249
- D. U.S. Patent Application 2003/0014441
- E. U.S. Patent 6,938,079

Response to Arguments

7. Applicant's arguments, filed on September 13, 2007, have been fully considered but are moot on new grounds of rejections.

Contact Information

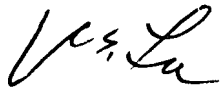
8. Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Kuen S. Lu whose telephone number is (571)-272-4114. The examiner can normally be reached on Monday-Friday (8:00 am-5:00 pm). If attempts to reach the examiner by telephone are unsuccessful, the examiner's Supervisor, John Cottingham can be reached on (571)-272-7079. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for Page 13 published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should

Art Unit: 2167

you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-27-9197 (toll free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, please call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Kuen S. Lu,



Patent Examiner, Art Unit 2167

November 21, 2007